Week 8: Activity Key

Last Week's Recap

- Exam 1
- Linear Algebra
- Linear Model Overview
- Simulating Data in R
- Fitting Linear Models in R
- Multivariate Normal distribution
- Partitioned Matrices and Conditional Multivariate normal distribution

Video Lectures

- `rstan` in R for Bayesian inference

This week

- Gaussian Process Intro
- Bayesian inference with `stan`
- Correlation functions

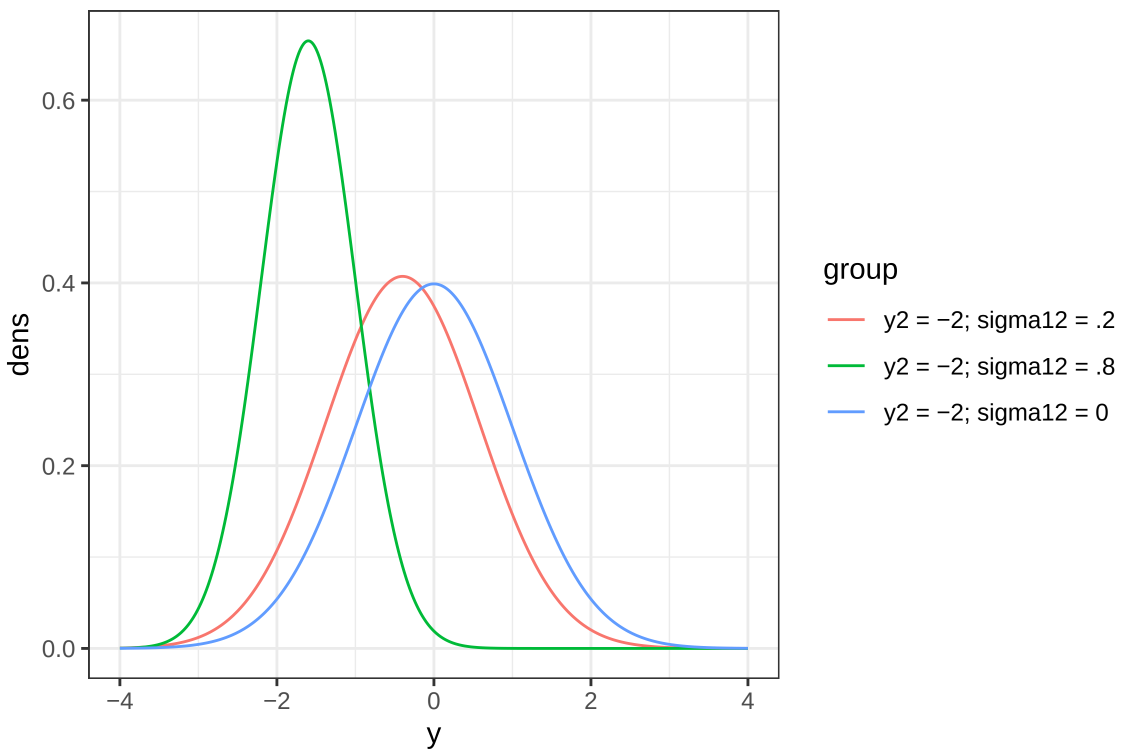
Visual Example

Let $n_1 = 1$ and $n_2 = 1$, then

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}\right)$$

and

$$y_1|y_2 \sim N\left(\mu_1 + \sigma_{12}(\sigma_2^2)^{-1}\left(y_2 - \mu_2\right), \sigma_1^2 - \sigma_{12}(\sigma_2^2)^{-1}\sigma_{21}\right)$$



Q: Calculate and write out the actual distributions for y_1 in these three settings.

1. $\sigma_{12} = 0 \rightarrow y_1|y_2 \sim N(0, 1^2)$
2. $\sigma_{12} = .2 \rightarrow y_1|y_2 \sim N(-.4, .96^2)$
3. $\sigma_{12} = .8 \rightarrow y_1|y_2 \sim N(-1.6, .36^2)$

One last note, the marginal distributions for any partition $\underline{y_1}$ are quite simple.

$$\underline{y_1} \sim N\left(X_1\beta, \Sigma_{11}\right)$$

or just

$$y_1 \sim N\left(X_1\beta, \sigma_1^2\right)$$

if y_1 is scalar.

GP Overview

Now let's extend this idea to a Gaussian Process (GP). There are two fundamental ideas to a GP.

1. Any finite set of realizations (say $\underline{y_2}$) has a multivariate normal distribution.

2. Conditional on a set of realizations, all other locations (say $\underline{y_1}$) have a conditional normal distribution characterized by the mean, and most importantly the covariance function. Note the dimension of $\underline{y_1}$ can actually be infinite, such as defined on the real line.

The big question is how to we estimate Σ_{12}? How many parameters are necessary for this distribution?

Generally, Σ_{12}, or more specifically the individual elements of Σ_{12}, such as $\sigma i, j$ will be estimated using some idea of distance.

Fundamental idea of spatial statistics is that things close together tend to be similar.

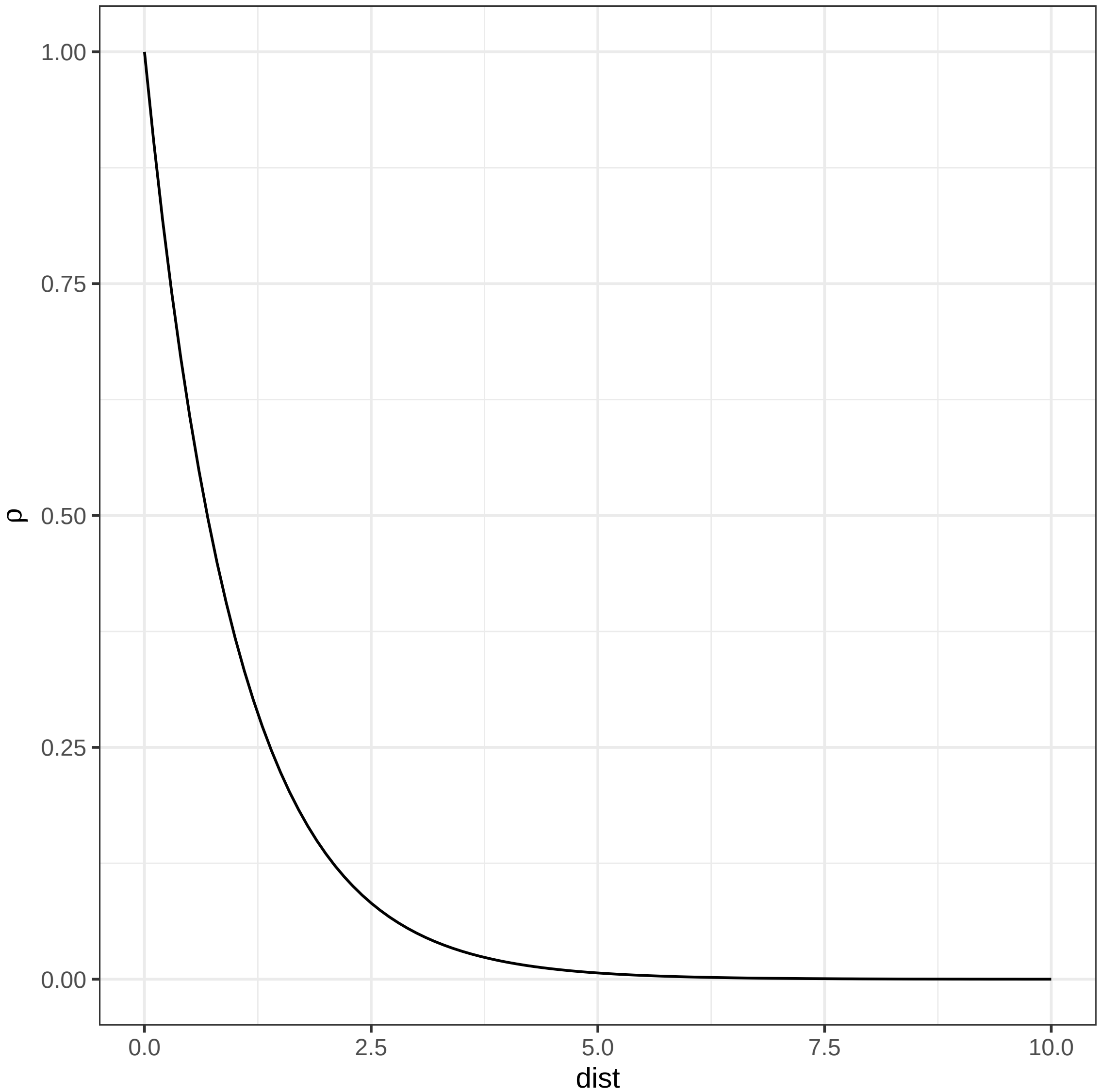
Correlation function

Initially, let's consider correlation as a function of distance, in one dimension or on a line.

As a starting point, consider a variant of what is known as the exponential covariance function - we used this earlier. First define d as the Euclidean distance between x_1 and x_2, such that $d = \sqrt{(x_i - x_j)^2}$

$$\rho_{i,j} = \exp\left(-d\right)$$

Create a figure that shows the exponential correlation as a function of distance between the two points.



Using a correlation function can reduce the number of unknown parameters in a covariance matrix. In an unrestricted case, Σ has $\binom{n}{2} + n$ unknown parameters. However, using a correlation function can reduce the number of unknown parameters substantially, generally less than 4.

Realizations of a Gaussian Process

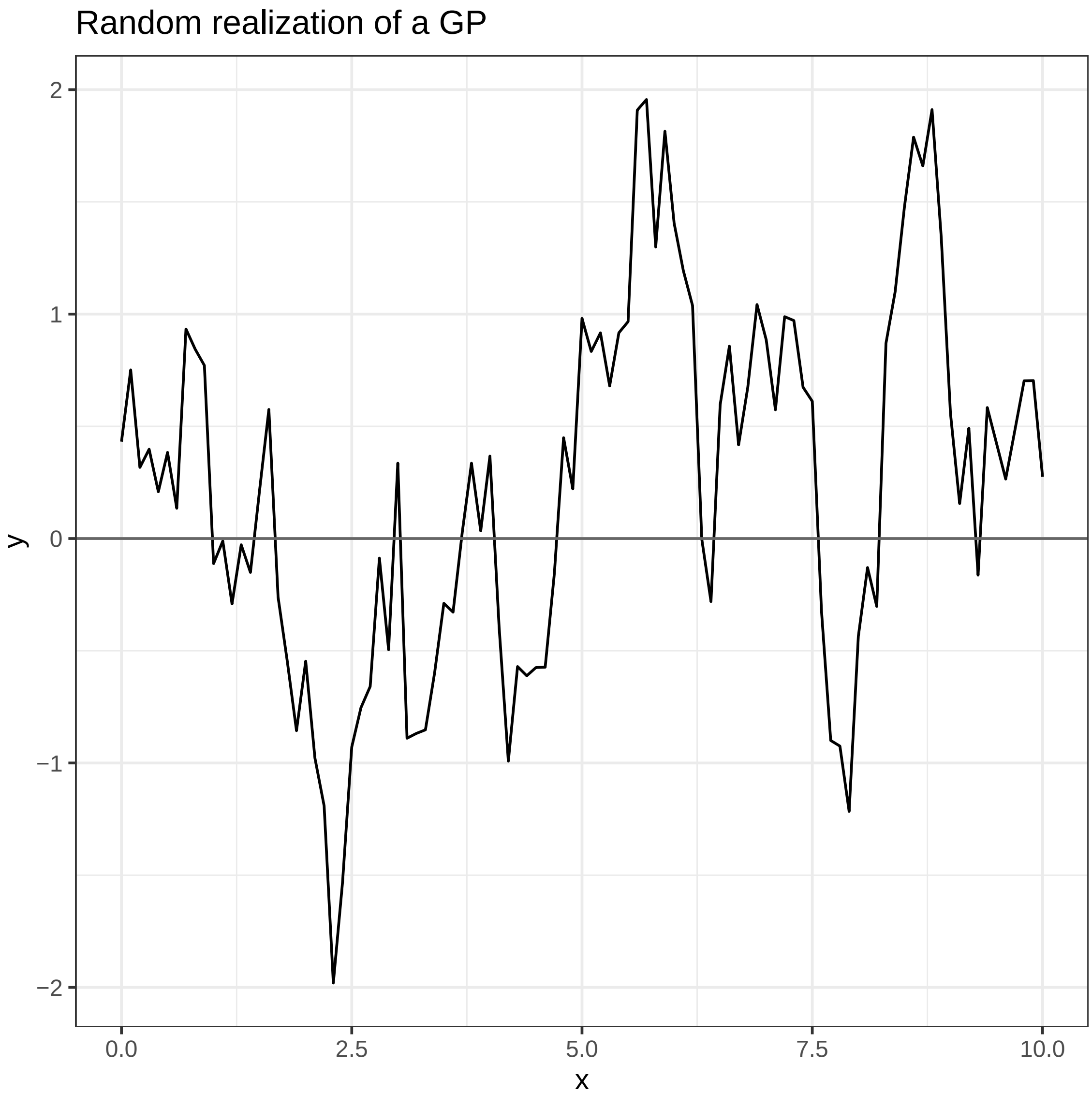
Recall that a process implies an infinite dimensional object. So we can generate a line rather than a discrete set of points. (While in practice the line will in fact be generated with a discrete set of points and then connected.)

For this scenario we will assume a zero-mean GP, with covariance equal to the correlation function using $\rho_{i,j} = \exp(-d)$

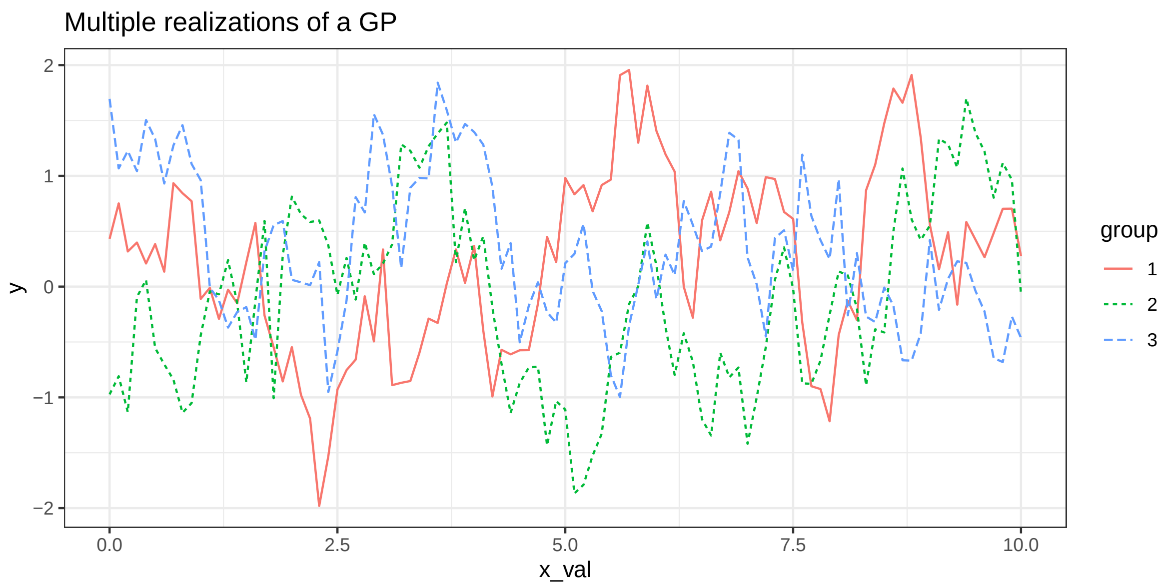
```
set.seed(02252025)
dist_mat <- as.matrix(dist(x, diag = T, upper = T))
Sigma <- exp(-dist_mat)

y <- rmnorm(n =1, mean = 0 , varcov = Sigma)

tibble(y = y, x = x) |>
  ggplot(aes(y=y, x=x)) +
  theme_bw() +
  geom_line() +
  geom_hline(yintercept = 0, color = 'grey40') +
  ggtitle('Random realization of a GP')
```

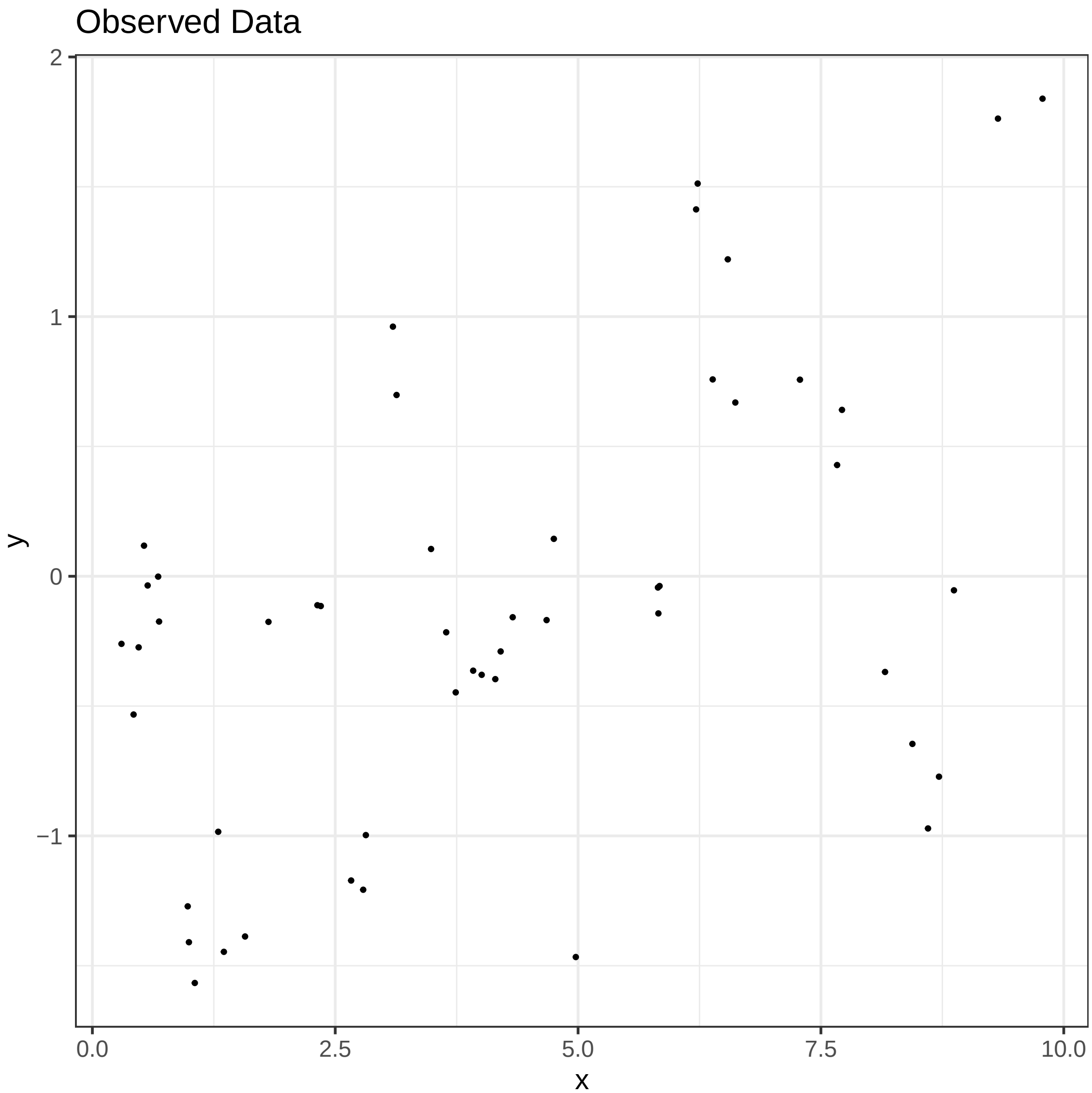


Overlay a few realizations of a Gaussian process on the same curve.


Multiple realizations of a GP
y
2
1
0
−1
−2
0.0
2.5
5.0
7.5
10.0
x_val
group
1
2
3

Connecting a GP to conditional normal

Now consider a discrete set of points, say $\underline{y_2}$, how can we estimate the response for the remainder of the values in the interval [0,10].



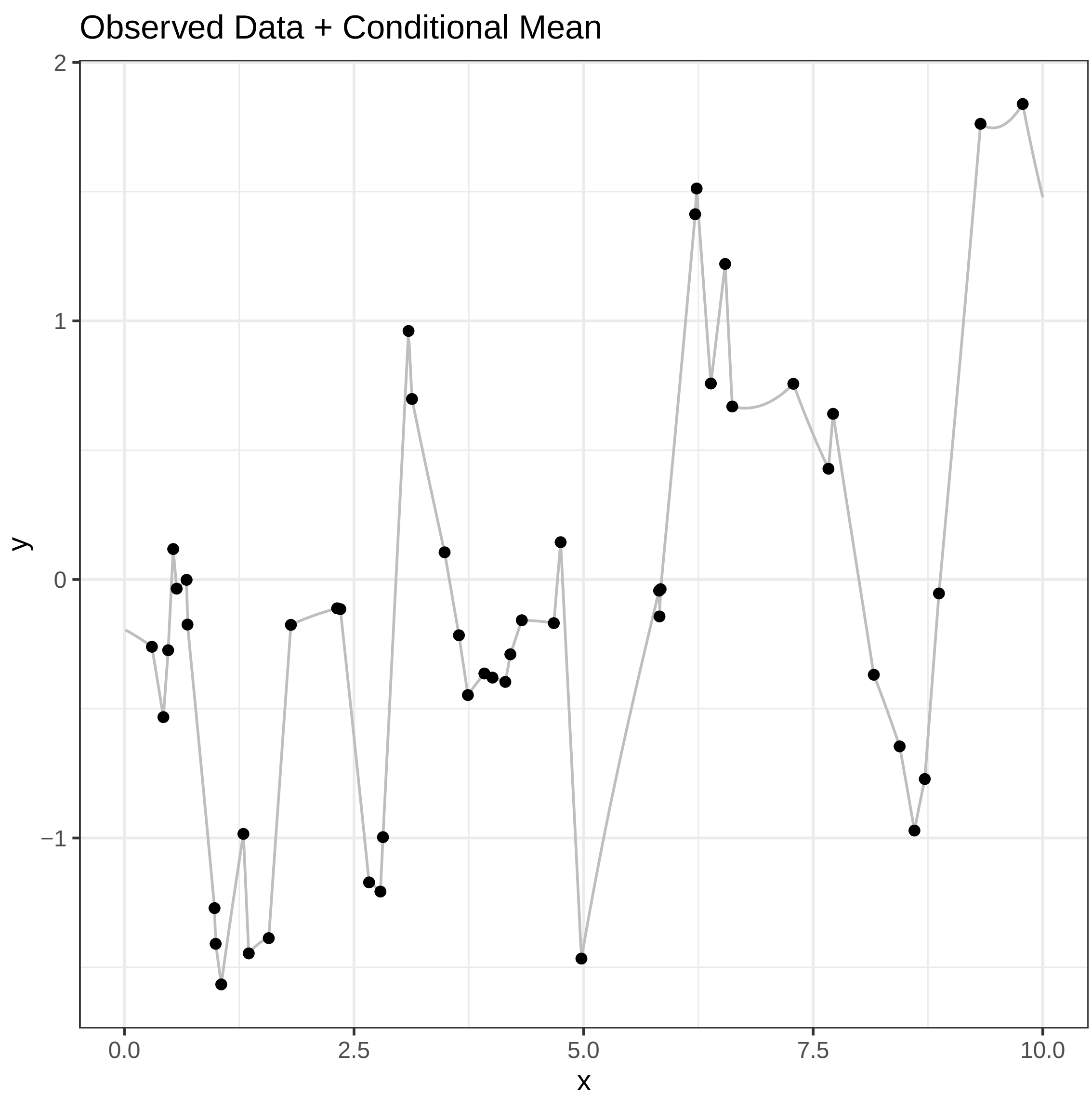
We can connect the dots (with uncertainty) using:

$$\underline{y_1}|\underline{y_2} \sim N\left(X_1\beta + \Sigma_{12}\Sigma_{22}^{-1}\left(\underline{y_2} - X_2\beta\right), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$$

Create a figure that shows the data points, conditional mean and uncertainty

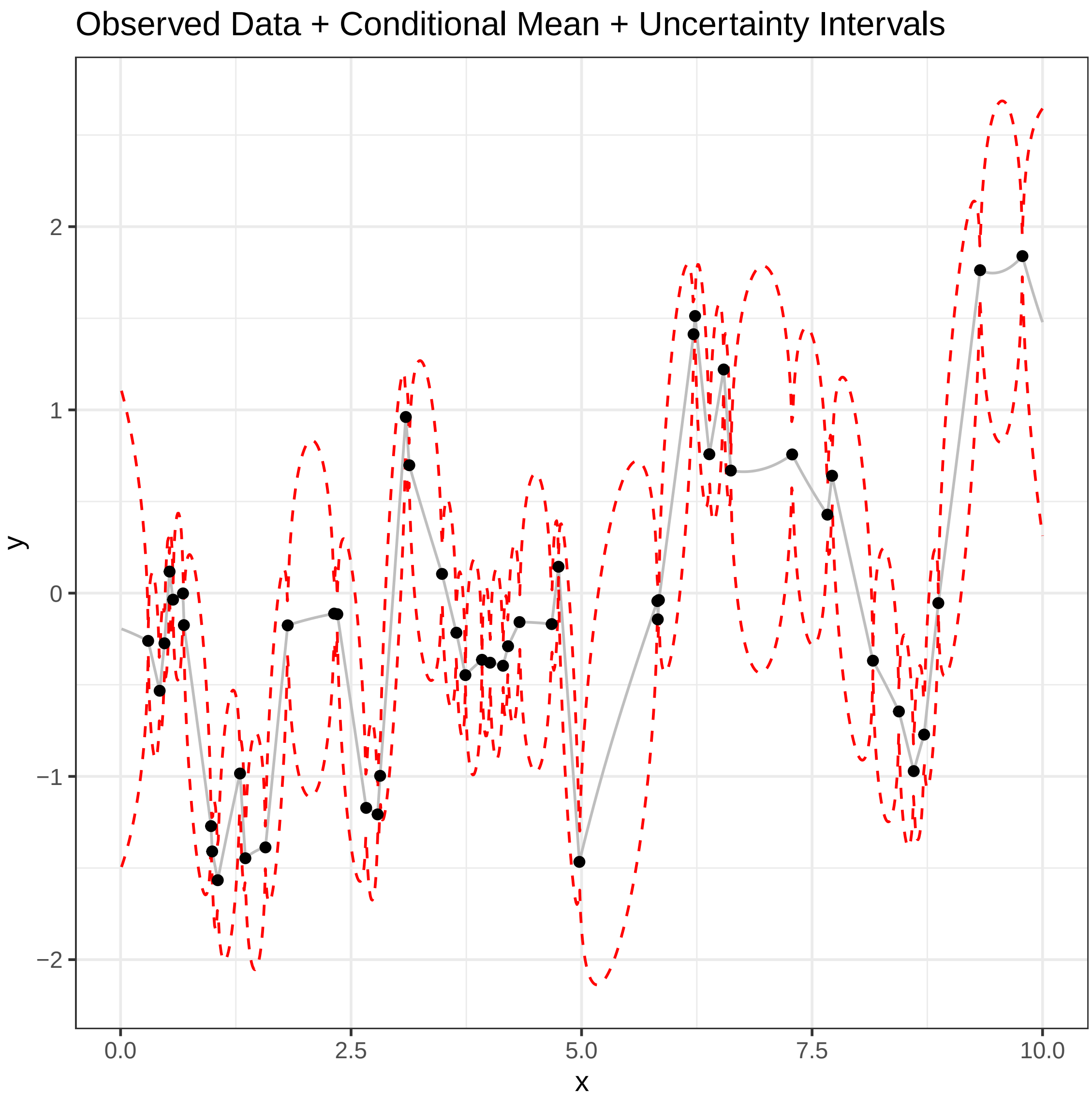
```
x1 <- seq(0.01, 10, .01)
n <- length(x1)
d1 <- as.matrix(dist(x1, diag = T, upper = T))
Sigma11 <- exp(-d1)
d12 <- sqrt(plgp::distance(x1,x2))
Sigma12 <- exp(-d12)
mu_1given2 <- Sigma12 %*% solve(Sigma22) %*% matrix(y2, nrow = length(y2), ncol = 1)
eps <- .Machine$double.eps

Sigma_1given2 <- Sigma11 - Sigma12 %*% solve(Sigma22) %*% t(Sigma12) +  diag(eps, n)
```




```
uncertainty_line <- tibble(y_mean = mu_1given2,
                    x1 = x1,
                    sd = sqrt(diag(Sigma_1given2) ),
                    upper = y_mean + 1.96 * sd,
                    lower = y_mean -1.96 * sd)
data_fig +
  geom_line(aes(y = y_mean, x = x1), inherit.aes = F, data = mean_line, color = 'gray') +
  geom_line(aes(y = upper, x = x1), inherit.aes = F, data = uncertainty_line, color = 'red',
  geom_line(aes(y = lower, x = x1), inherit.aes = F, data = uncertainty_line, color = 'red',
```

```
  geom_point() +   ggtitle('Observed Data + Conditional Mean + Uncertainty Intervals')
```

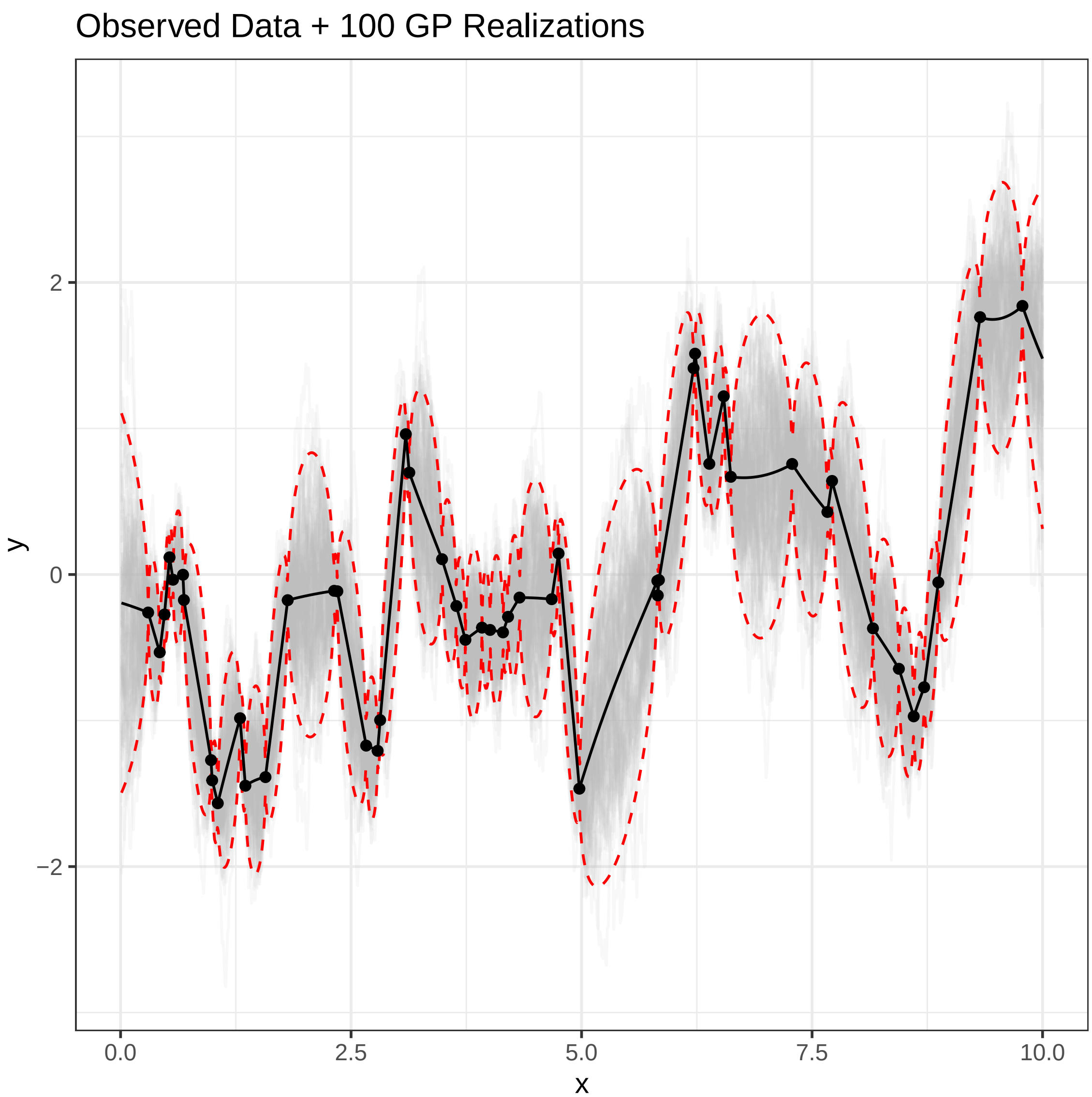



```
num_sims <- 100
y1_sims <- rmnorm(num_sims, mu_1given2, Sigma_1given2)

long_sims <- y1_sims %>% melt() %>% bind_cols(tibble(x = rep(x1, each = num_sims)))

data_and_mean +
  geom_line(aes(y = value, x = x, group = Var1), inherit.aes = F,
```

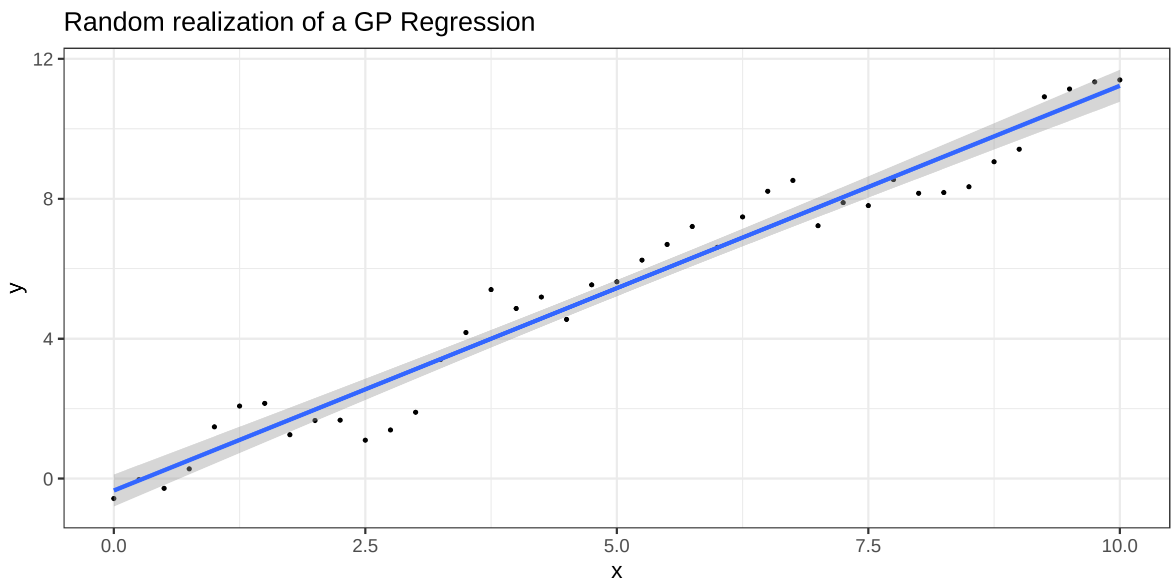
```
          data = long_sims, alpha = .1, color = 'gray') +
  ggtitle('Observed Data + 100 GP Realizations') +
  geom_line(aes(y = upper, x = x1), inherit.aes = F, data = uncertainty_line, color = 'red',
  geom_line(aes(y = lower, x = x1), inherit.aes = F, data = uncertainty_line, color = 'red',
  geom_line(aes(y = y_mean, x = x1), inherit.aes = F, data = mean_line, color = 'black') +
  geom_point(data = tibble(y = y2, x = x2))
```



GP Regression

Now rather than specifying a zero-mean GP, let the mean be $X\underline{\beta}$.

```
x <- seq(0, 10, by = .25)
beta <- 1
n <- length(x)
d <- sqrt(plgp::distance(x))
H <- exp(-d)
y <- rmnorm(1, x * beta ,H)
```

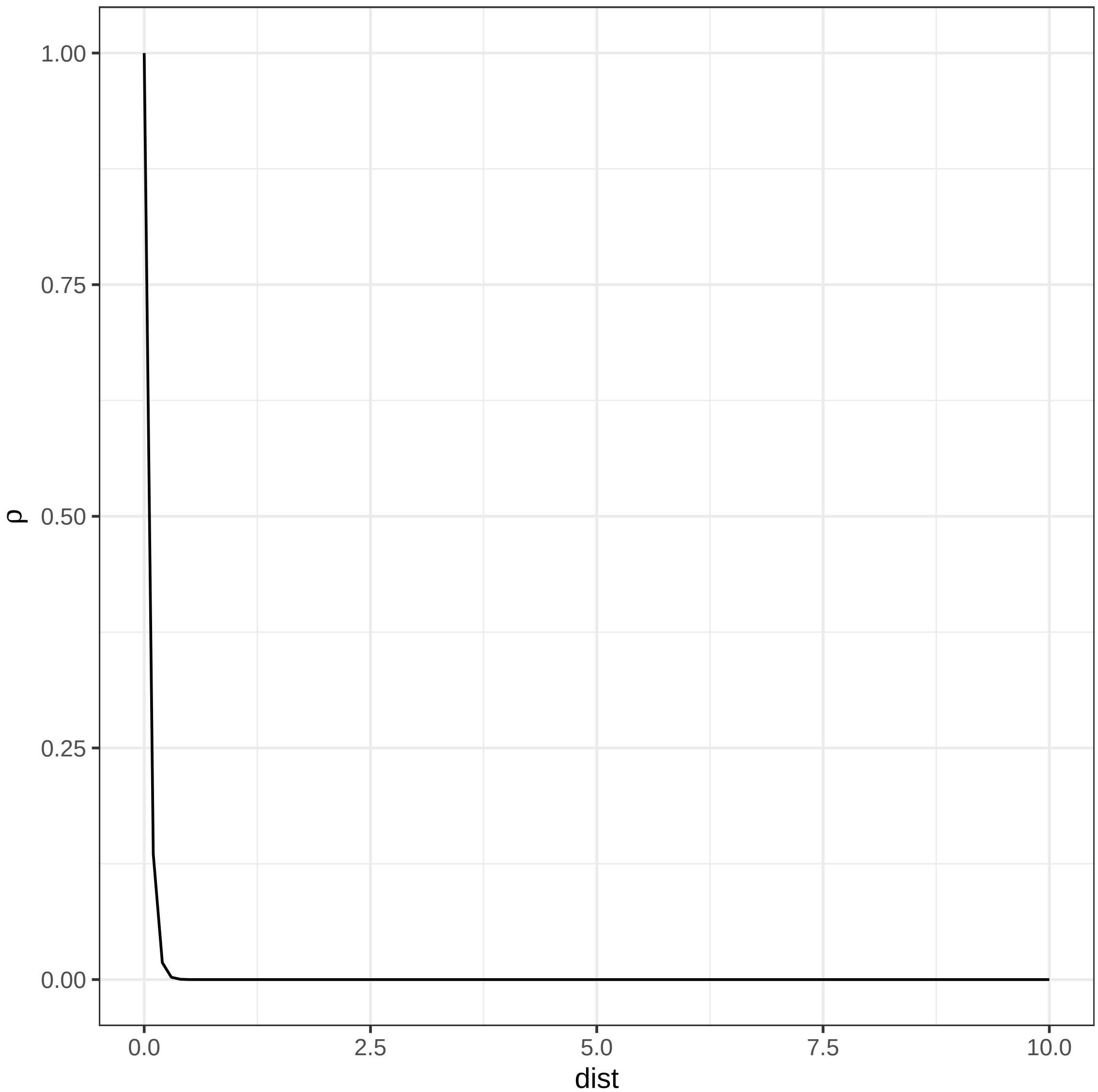


Correlation function: more details

Recall the variant of the exponential covariance function that we have previously seen. Where d as the Euclidean distance between x_1 and x_2, such that $d = \sqrt{(x_i - x_j)^2}$

$$\rho_{i,j} = \exp\left(-d\right)$$

Lets view the exponential correlation as a function of distance between the two points.



Now let's consider a more general framework for covariance where

$$\sigma_{i,j} = \sigma^2 \exp\left(-d_{ij}/\phi\right)$$

Now we have introduced two new parameters into this function. What do you suppose that they do?

- σ^2: controls the magnitude of the covariance.

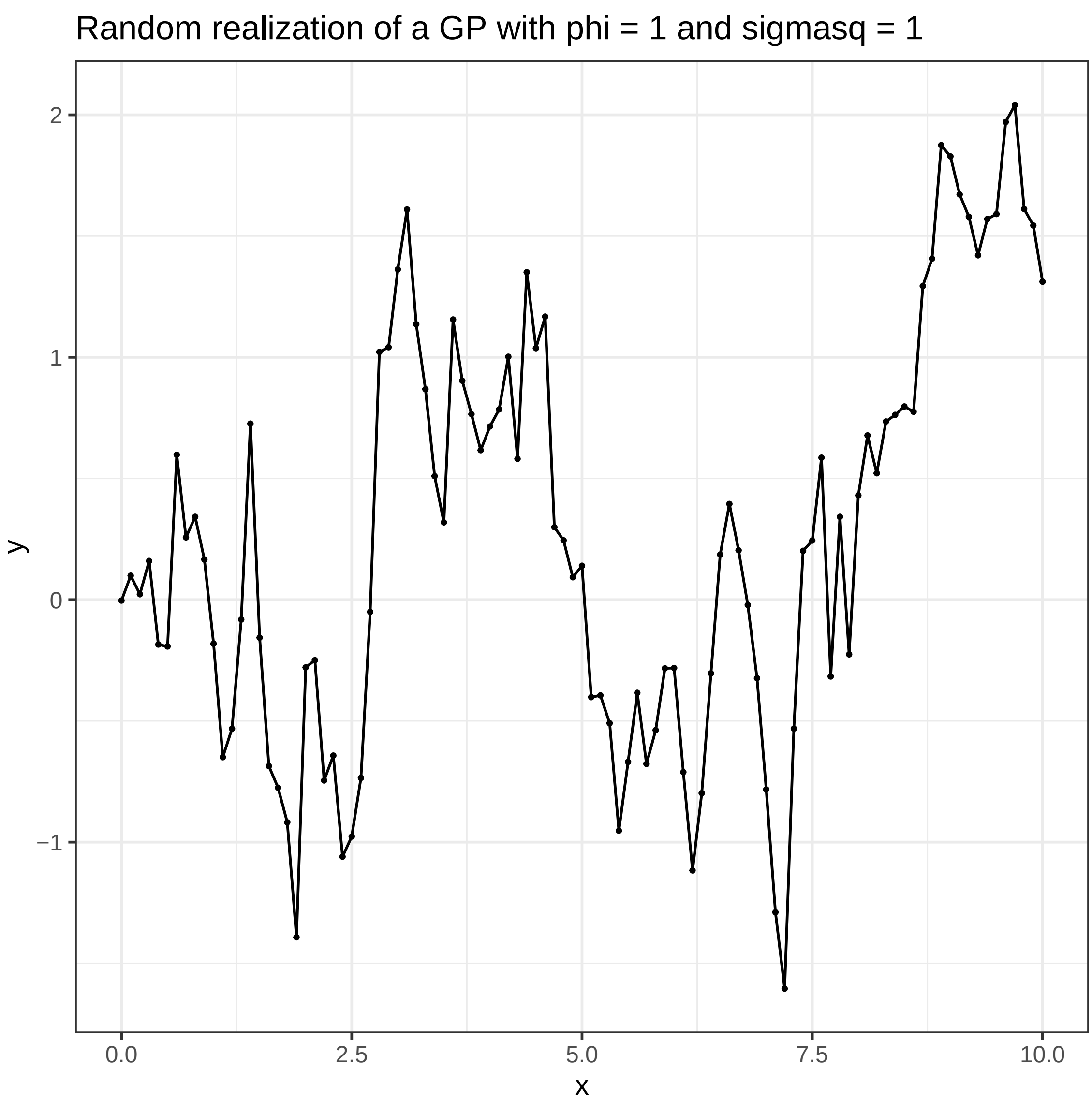
- ϕ: controls the range of the spatial correlation

Modify your previous code do adjust ϕ and σ^2 and explore how they differ.

```r
phi <- 1
sigmasq <- 1
x <- seq(0, 10, by = .1)
n <- length(x)
d <- sqrt(plgp::distance(x))
eps <- sqrt(.Machine$double.eps)
H <- exp(-d/phi) + diag(eps, n)
H[1:3,1:3]
```

```
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.9048374 0.8187308
[2,] 0.9048374 1.0000000 0.9048374
[3,] 0.8187308 0.9048374 1.0000000
```

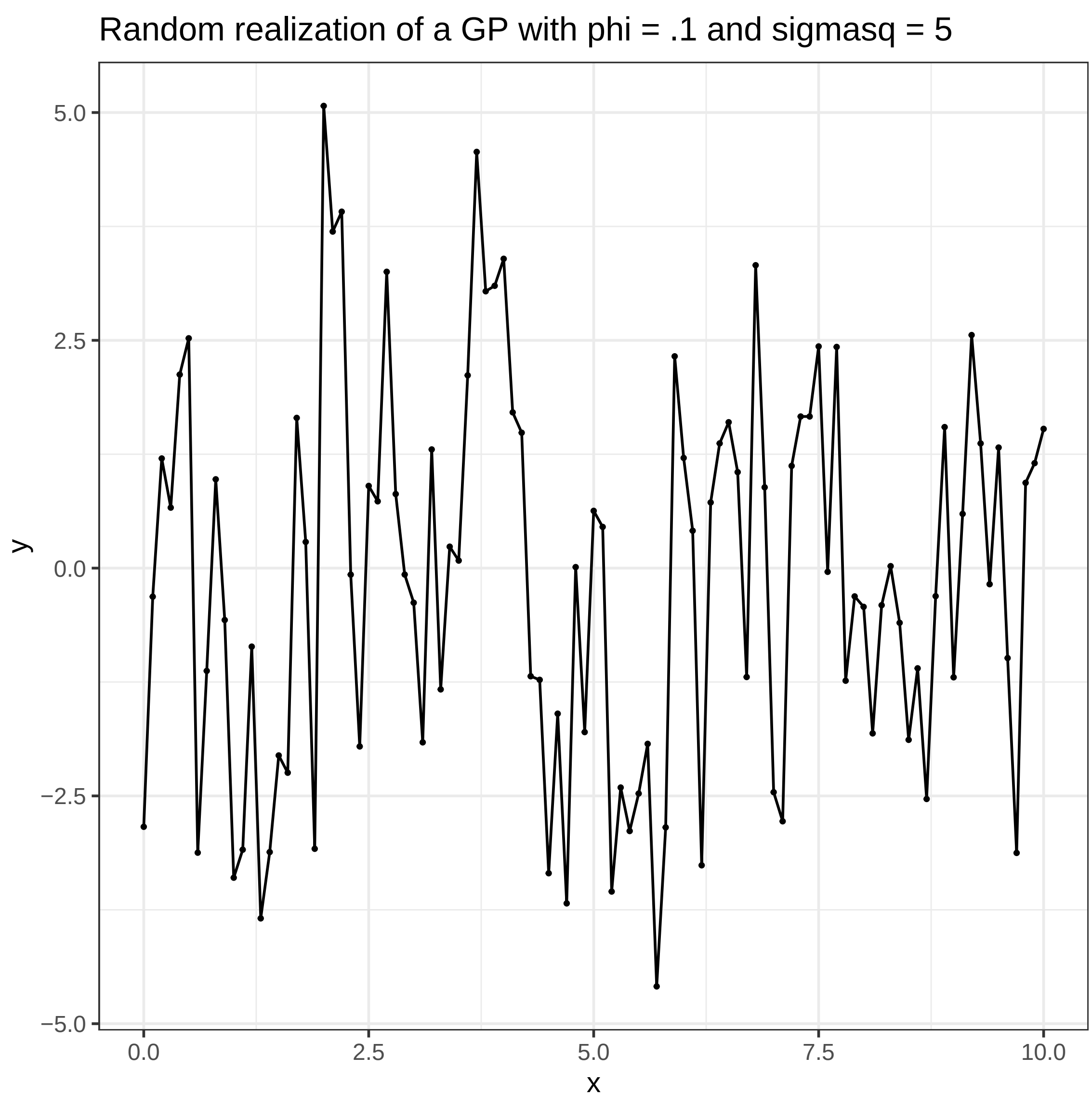
```
y <- rmnorm(1, rep(0,n),sigmasq * H)
tibble(y = y, x = x) %>% ggplot(aes(y=y, x=x)) +
  geom_line() + theme_bw() + ggtitle('Random realization of a GP with phi = 1 and sigmasq = 1
  geom_point(size = .5)
```




```
phi <- .1
sigmasq <- 5
H <- exp(-d/phi) + diag(eps, n)
H[1:3,1:3]
```

```
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.3678794 0.1353353
[2,] 0.3678794 1.0000000 0.3678794
[3,] 0.1353353 0.3678794 1.0000000
```

```
y <- rmnorm(1, rep(0,n),sigmasq * H)
tibble(y = y, x = x) %>% ggplot(aes(y=y, x=x)) +
  geom_line() + theme_bw() + ggtitle('Random realization of a GP with phi = .1 and sigmasq =
  geom_point(size = .5)
```



We will soon talk about a more broad set of correlation functions and another parameter that provides flexibility so that predictions do not have to directly through observed points.

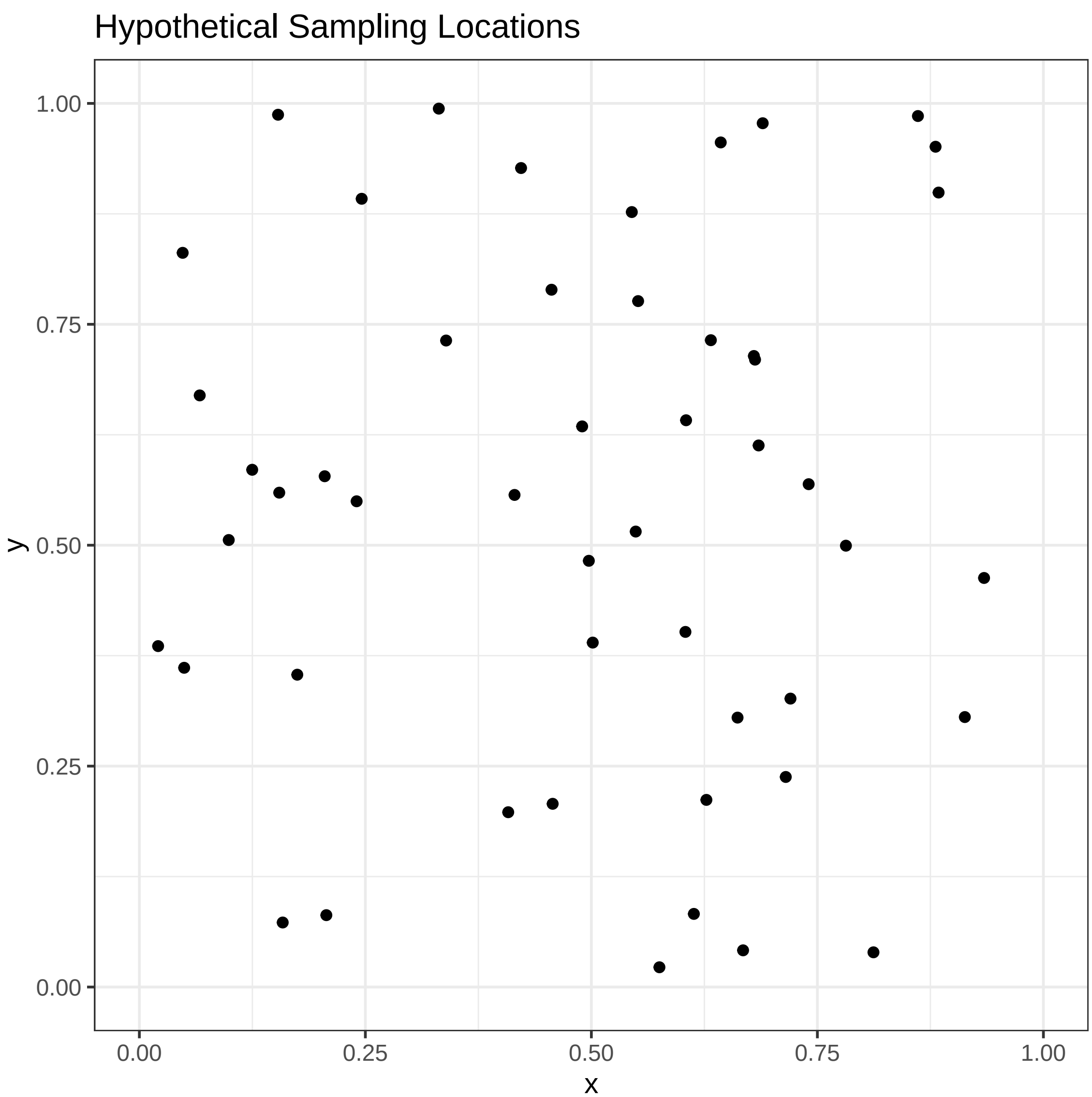
Geostatistical Data

At last, we will look at simulated 2-d “spatial” data.

1. Create Sampling Locations

```
set.seed(03062025)

num.locations <- 50
coords <- data.frame(x = runif(num.locations), y = runif(num.locations))
coords %>% ggplot(aes(x=x,y=y)) + geom_point() +
  ggtitle('Hypothetical Sampling Locations') + xlim(0,1) +
  ylim(0,1) + theme_bw()
```



2. Calculate Distances

```
dist.mat <- sqrt(plgp::distance(coords))
```

3. Define Covariance Function and Set Parameters

Use exponential covariance with no nugget:

```
sigma.sq <- 1
phi <- .1
Sigma <- sigma.sq * exp(- dist.mat/phi) + diag(eps, num.locations)
```

4. Sample realization of the process

- This requires a distributional assumption, we will use the Gaussian distribution

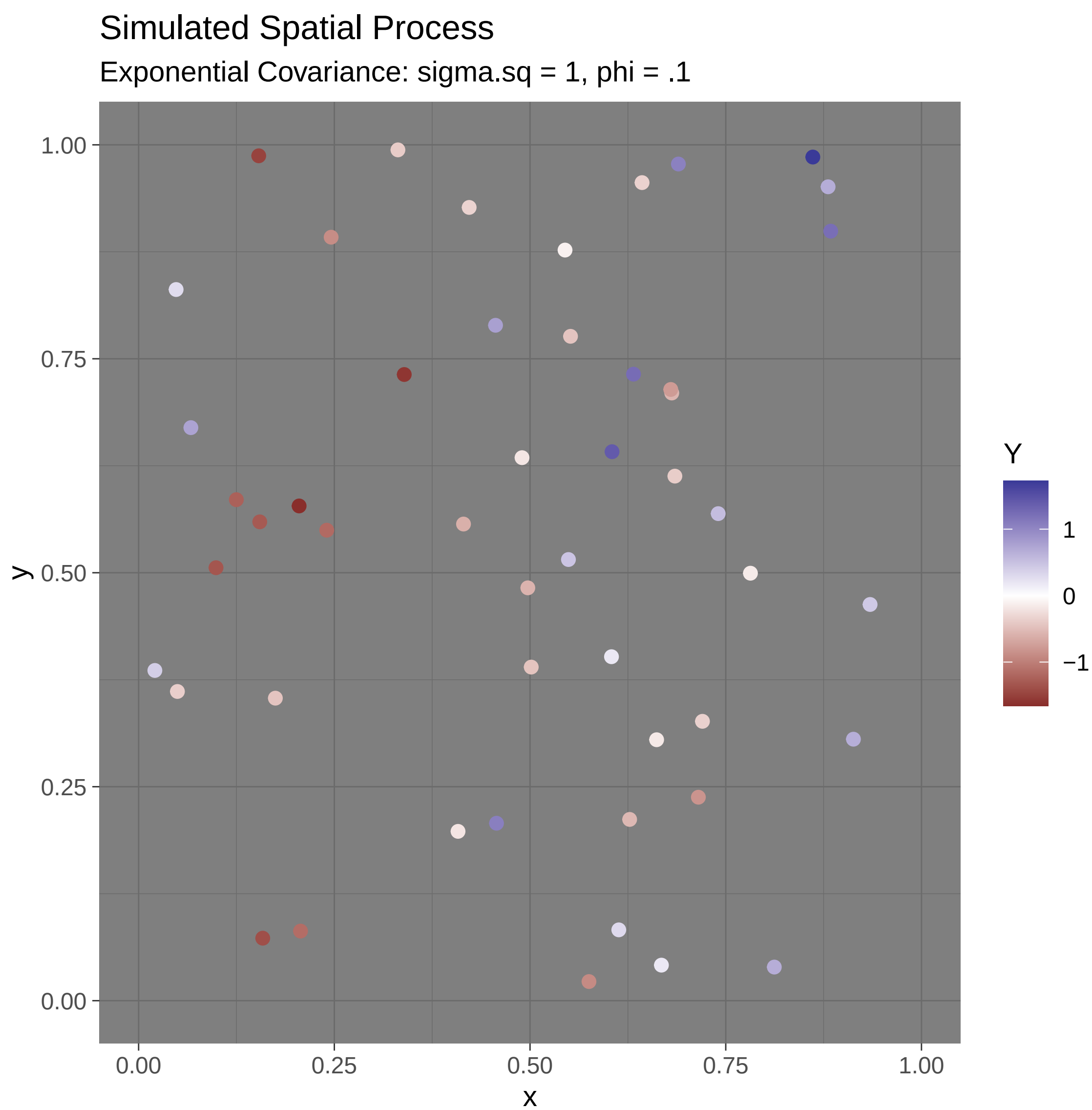
```
Y <- rmnorm(n=1, mean = 0, varcov = Sigma)
```

- What about the rest of the locations on the map?

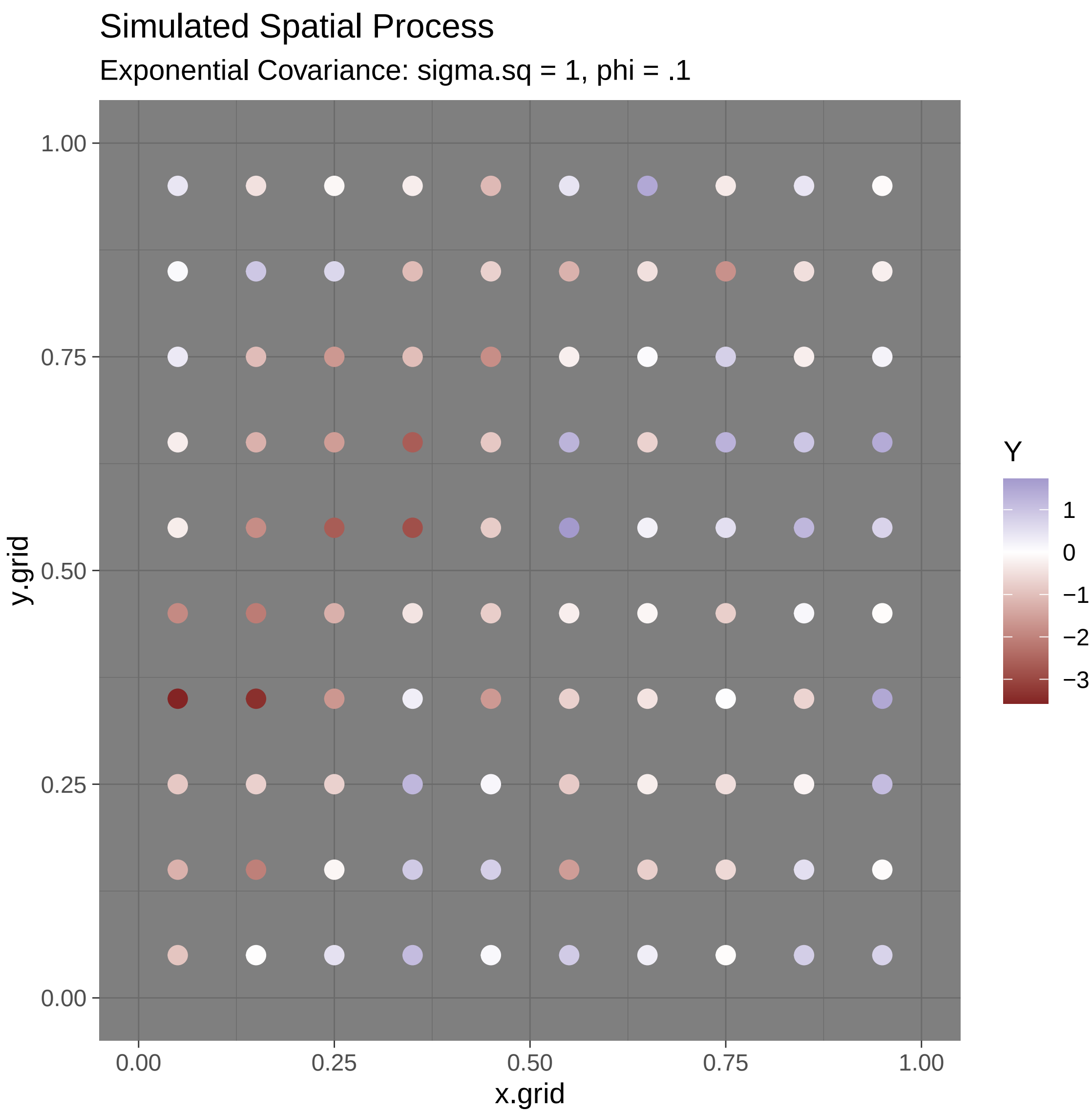
5. Vizualize Spatial Process

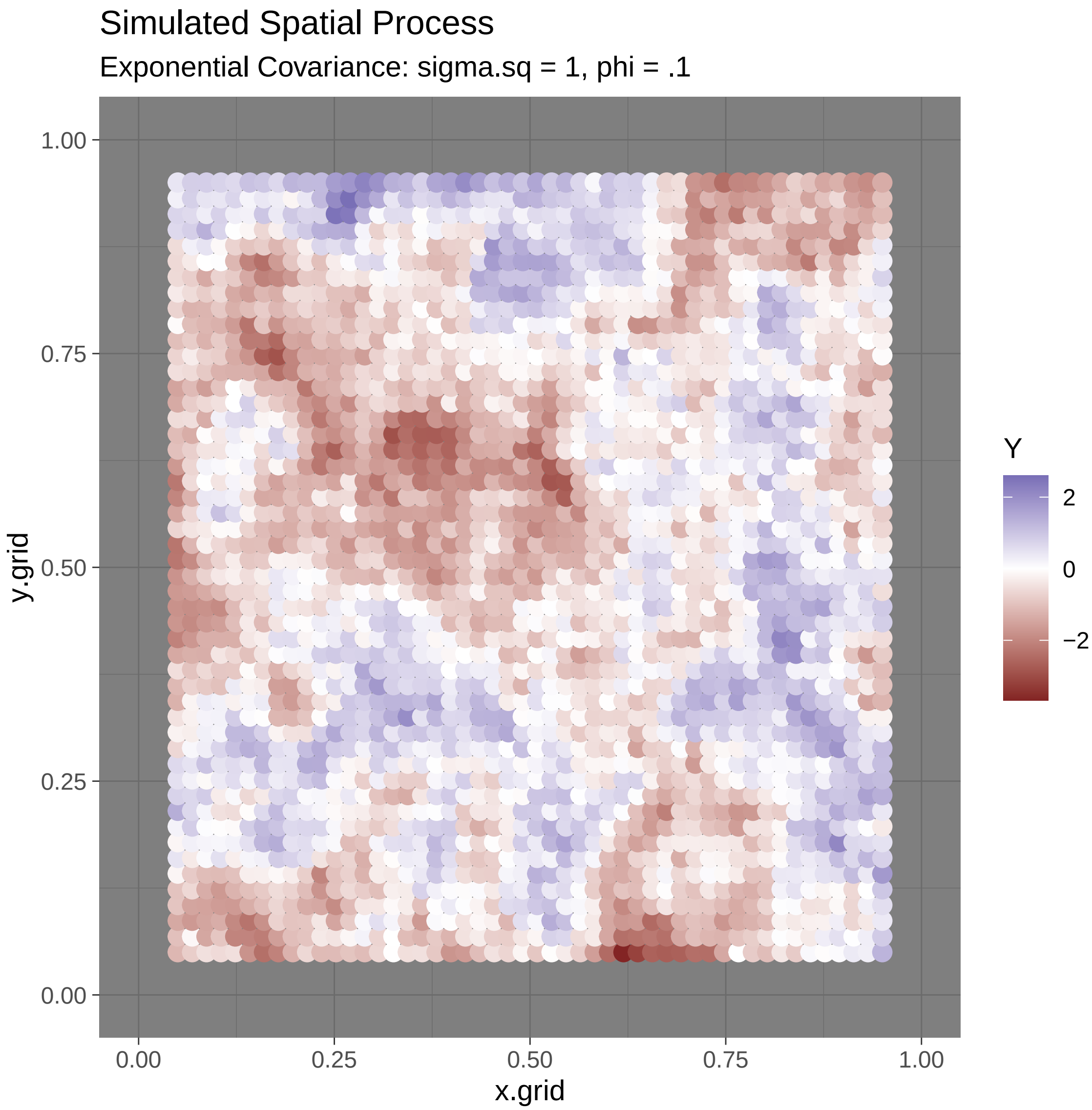
Start with a coarse grid and them move to a finer grid

```
coords %>% mutate(Y = Y) %>% ggplot(aes(x=x,y=y)) + geom_point(aes(color=Y), size=2) +
  ggtitle(label = 'Simulated Spatial Process',
          subtitle = 'Exponential Covariance: sigma.sq = 1, phi = .1') +
  xlim(0,1) + ylim(0,1) +   scale_colour_gradient2() + theme_dark()
```


Simulated Spatial Process
Exponential Covariance: sigma.sq = 1, phi = .1
y
1.00
0.75
0.50
0.25
0.00
x
0.00
0.25
0.50
0.75
1.00
Y
1
0
−1

Now we can look at more sampling locations





How does the spatial process change with:

- another draw with same parameters?
- a different value of ϕ
- a different value of σ^2